ClickOnce Deployment and Security - Visual Studio (Windows)

By Mikejo5000

Archived: 2026-04-05 20:24:44 UTC

ClickOnce is a deployment technology that enables you to create self-updating Windows-based applications that can be installed and run with minimal user interaction. Visual Studio provides full support for publishing and updating applications deployed with ClickOnce technology if you have developed your projects with Visual Basic and Visual C#. For information about deploying Visual C++ applications, see ClickOnce Deployment for Visual C++ Applications.

ClickOnce deployment overcomes three major issues in deployment:

- **Difficulties in updating applications.** With Microsoft Windows Installer deployment, whenever an application is updated, the user can install an update, an msp file, and apply it to the installed product; with ClickOnce deployment, you can provide updates automatically. Only those parts of the application that have changed are downloaded, and then the full, updated application is reinstalled from a new side-by-side folder.

- **Impact to the user's computer.** With Windows Installer deployment, applications often rely on shared components, with the potential for versioning conflicts; with ClickOnce deployment, each application is self-contained and can't interfere with other applications.

- **Security permissions.** Windows Installer deployment requires administrative permissions and allows only limited user installation; ClickOnce deployment enables non-administrative users to install and grants only those Code Access Security permissions necessary for the application.

  In the past, these issues sometimes caused developers to decide to create Web applications instead of Windows-based applications, sacrificing a rich user interface for ease of installation. By using applications deployed using ClickOnce, you can have the best of both technologies.

What is a ClickOnce application?

A ClickOnce application is any Windows Presentation Foundation (*.xbap*), Windows Forms (*.exe*), console application (*.exe*), or Office solution (*.dll*) published using ClickOnce technology. You can publish a ClickOnce application in three different ways: from a Web page, from a network file share, or from legacy media such as a CD-ROM. A ClickOnce application can be installed on an end user's computer and run locally even when the computer is offline, or it can be run in an online-only mode without permanently installing anything on the end user's computer. For more information, see Choose a ClickOnce deployment strategy.

ClickOnce applications can be self-updating; they can check for newer versions as they become available and automatically replace any updated files. The developer can specify the update behavior; a network administrator can also control update strategies, for example, marking an update as mandatory. Updates can also be rolled back

to an earlier version by the end user or by an administrator. For more information, see Choose a ClickOnce update strategy.

Because ClickOnce applications are isolated, installing or running a ClickOnce application can't break existing applications. ClickOnce applications are self-contained; each ClickOnce application is installed to and run from a secure per-user, per-application cache. ClickOnce applications run in the Internet or Intranet security zones. If necessary, the application can request elevated security permissions. For more information, see Secure ClickOnce applications.

How ClickOnce security works

The core ClickOnce security is based on certificates, code access security policies, and the ClickOnce trust prompt.

Certificates

Authenticode certificates are used to verify the authenticity of the application's publisher. By using Authenticode for application deployment, ClickOnce helps prevent a harmful program from portraying itself as a legitimate program coming from an established, trustworthy source. Optionally, certificates can also be used to sign the application and deployment manifests to prove that the files haven't been tampered with. For more information, see ClickOnce and Authenticode. Certificates can also be used to configure client computers to have a list of trusted publishers. If an application comes from a trusted publisher, it can be installed without any user interaction. For more information, see Trusted application deployment overview.

Code access security

Code access security helps limit the access that code has to protected resources. In most cases, you can choose the Internet or Local Intranet zones to limit the permissions. Use the **Security** page in the **Project Designer** to request the zone appropriate for the application. You can also debug applications with restricted permissions to emulate the end-user experience. For more information, see Code access security for ClickOnce applications.

Note

In ClickOnce for .NET Core and .NET 5 or later, Code Access Security is unsupported. In .NET Framework, the use of Code Access Security is not a best practice and is not recommended.

ClickOnce trust prompt

If the application requests more permissions than the zone allows, the end user can be prompted to make a trust decision. The end user can decide if ClickOnce applications such as Windows Forms applications, Windows Presentation Foundation applications, console applications, XAML browser applications, and Office solutions are trusted to run. For more information, see How to: Configure the ClickOnce trust prompt behavior.

How ClickOnce deployment works

The core ClickOnce deployment architecture is based on two XML manifest files: an application manifest and a deployment manifest. The files are used to describe where the ClickOnce applications are installed from, how they're updated, and when they're updated.

Publish ClickOnce applications

The application manifest describes the application itself. This includes the assemblies, the dependencies and files that make up the application, the required permissions, and the location where updates will be available. The application developer authors the application manifest by using the Publish Wizard in Visual Studio (Publish tool for .NET Core and .NET 5+) or the Manifest Generation and Editing Tool (*Mage.exe*) in the Windows Software Development Kit (SDK). For more information, see:

- Deploy a .NET desktop application using ClickOnce
- Deploy a .NET Framework desktop application using ClickOnce.

The deployment manifest describes how the application is deployed. This includes the location of the application manifest, and the version of the application that clients should run.

Note

In ClickOnce for .NET Core 3.1 and .NET 5 or later, use *dotnet-mage.exe* instead of *Mage.exe*. For more information, see ClickOnce for .NET.

Deploy ClickOnce applications

After it's created, the deployment manifest is copied to the deployment location. This can be a Web server, network file share, or legacy media such as a CD. The application manifest and all the application files are also copied to a deployment location that is specified in the deployment manifest. This can be the same as the deployment location, or it can be a different location. When using the **Publish Wizard** in Visual Studio, the copy operations are performed automatically.

Install ClickOnce applications

After it's deployed to the deployment location, end users can download and install the application by clicking an icon representing the deployment manifest file on a Web page or in a folder. In most cases, the end user is presented with a simple dialog box asking the user to confirm installation, after which installation proceeds and the application is started without additional intervention. In cases where the application requires elevated permissions or if the application isn't signed by a trusted certificate, the dialog box also asks the user to grant permission before the installation can continue. Though ClickOnce installs are per-user, permission elevation may be required if there are prerequisites that require administrator privileges. For more information about elevated permissions, see Securing ClickOnce applications.

Certificates can be trusted at the machine or enterprise level, so that ClickOnce applications signed with a trusted certificate can install silently. For more information about trusted certificates, see Trusted application deployment overview.

The application can be added to the user's **Start** menu and to the **Add or Remove Programs** group in the **Control Panel**. Unlike other deployment technologies, nothing is added to the **Program Files** folder or the registry, and no administrative rights are required for installation

Note

It's also possible to prevent the application from being added to the **Start** menu and **Add or Remove Programs** group, in effect making it behave like a Web application. For more information, see Choose a ClickOnce deployment strategy.

Update ClickOnce applications

When the application developers create an updated version of the application, they generate a new application manifest and copy files to a deployment location—usually a sibling folder to the original application deployment folder. The administrator updates the deployment manifest to point to the location of the new version of the application.

Note

The **Publish Wizard** in Visual Studio can be used to perform these steps. For .NET Core and .NET 5+, the Publish tool provides these steps.

In addition to the deployment location, the deployment manifest also contains an update location (a Web page or network file share) where the application checks for updated versions. ClickOnce **Publish** properties are used to specify when and how often the application should check for updates. Update behavior can be specified in the deployment manifest, or it can be presented as user choices in the application's user interface by means of the ClickOnce APIs. In addition, **Publish** properties can be employed to make updates mandatory or to roll back to an earlier version. For more information, see Choosing a ClickOnce update strategy.

Third party installers

You can customize your ClickOnce installer to install third-party components along with your application. You must have the redistributable package (.exe or .msi file) and describe the package with a language-neutral product manifest and a language-specific package manifest. For more information, see Creating bootstrapper packages.

The following table shows the tools that you can use to generate, edit, sign, and re-sign the application and deployment manifests. For .NET Core and .NET 5+, options similar to MSBuild attributes are set using the Publish profile.

The following table shows the .NET Framework version required to support ClickOnce applications in these browsers.

Browser	.NET Framework version
Firefox	2.0 SP1, 3.5 SP1, 4
Chrome	3.5

Browser	.NET Framework version
Microsoft Edge	3.5

Related content

- Publish ClickOnce applications
- Secure ClickOnce applications
- Deploy COM components with ClickOnce
- Build ClickOnce applications from the command line
- Debug ClickOnce applications that use System.Deployment.Application
- ClickOnce deployment on older versions of Windows
